XRBars: Fast and Reliable Multiple Choice Selection by Gaze in XR

João Vitor Nogueira
Federal Fluminense University
Rio das Ostras, Rio de Janeiro, Brazil
jv_nogueira@id.uff.br

Carlos Morimoto
University of São Paulo
São Paulo, São Paulo, Brazil
hitoshi@ime.usp.br

Abstract

Web browsing is essential for modern education, supporting everything from self-directed study to academic research. However, traditional web interfaces are designed for keyboards and mice, and touch screens creating accessibility barriers for users with motor impairments. Modern Head-Mounted Displays (HMDs), which lack these conventional input devices but often include eye-tracking technology, make gaze-based interaction in XR a promising alternative due to its immersive experience. For a gaze-based XR web browser to be viable, it must be efficient and provide a good user experience. To this end, this paper proposes `XRBars`, a system that leverages GazeBars to improve accessibility to online educational resources in XR.

Keywords

Gaze Interaction, Web Browsing, Mixed Reality

How to cite this paper:
João Vitor Nogueira and Carlos Morimoto. 2025. XRBars: Fast and Reliable Multiple Choice Selection by Gaze in XR. In *Proceedings of ACM IMX Workshops, June 3 - 6, 2025*. SBC, Porto Alegre/RS, Brazil,5 pages. https://doi.org/10.5753/imxw.2025.2085

1 Introduction

Since its inception and widespread adoption, the web has fundamentally transformed how various processes are conducted. Access to information, public and private services, and essential tools has increasingly shifted online, making the internet a central hub for communication, work, and education [10].

As an educational tool, the web supports research and information retrieval, serves as a teaching platform, facilitates discussions through educational forums, bridges informal and formal learning, and functions as an assessment tool, among other applications. Its interconnected nature fosters collaborative learning, while its diverse multimedia resources provide extensive opportunities for knowledge acquisition [1, 19, 30].

While web-based education has significantly improved access to digital resources, it has also introduced barriers for users who struggle to navigate online platforms effectively. Many web-based learning systems still lack accessibility and often fail to comply with Web Content Accessibility Guidelines (WCAG), leaving individuals with physical disabilities unable to fully engage with educational content due to their inability to operate more traditional devices [5, 6].

Emerging technologies, such as Head-Mounted Displays (HMDs), expand the possibilities for accessing web-based learning by enabling alternative interaction methods that do not rely on traditional input devices. As enablers of Extended Reality (XR), HMDs provide immersive environments where users can interact with digital content in new ways. While still in its early stages, Web-XR has demonstrated potential for tasks such as assembly guidance, 3D visualization, and immersive collaborative learning [4, 11, 21, 33].

HMDs integrate a wide range of sensors, allowing for alternative input methods such as voice commands, gesture-based controls, head-pointing, gaze interaction, and multimodal approaches [36]. Among these, gaze-based interaction is particularly well-suited for web browsing, as it aligns with the web's inherent reliance on pointing and selection tasks. Unlike voice commands, which struggle with precise navigation, or hand gestures, which can be fatiguing, gaze-based controls provide a discrete and efficient means of interaction. Furthermore, gaze direction is closely linked to user intent, making it a valuable source of contextual information for user interfaces.

Gaze interaction has been extensively studied in desktop environments but stands to benefit from HMD integration due to the increasing affordability and widespread adoption of devices such as Apple's Vision Pro and Microsoft's HoloLens. These devices, which include built-in eye-tracking technology, are comparable in cost to standalone eye trackers but are standalone system, making them more accessible, and can be used in a wider variety of locations since they are wearable devices [12].

However, gaze-based interaction presents several challenges. Unlike more traditional interfaces, gaze lacks an explicit selection mechanism, requiring alternative methods to confirm user intent. Additionally, involuntary eye movements, such as glances or exploratory gazes, can result in unintended selections [20]. Precision limitations in current eye-tracking technology can also make small target selection difficult.

To address these challenges, various design solutions have been proposed. Most gaze-based systems employ dwell-time selection, where users must fixate on an interface element for a predefined duration (typically 500–1000ms) to confirm their choice. Assistive techniques such as target enlargement, selection timers, and magnification are commonly used to enhance accuracy. However, dwell-time selection has drawbacks: if the required duration is too short, it increases the risk of accidental selections, whereas longer dwell times can slow interactions and degrade user experience.

As an alternative to dwell-time selection, this paper presents the design of `XRBars`, a dwell-free web browser developed as part of an ongoing project. `XRBars` leverages bar placement and instantaneous

https://doi.org/10.5753/imxw.2025.2085

selection through GazeBars to enhance interaction speed and user experience in gaze-based web navigation.

2 Related Works

2.1 Gaze Interaction for the Web

Efforts in gaze-based browsers have been focused mainly on element selection and text entry. Those two features are key to web-interaction as text-entry allows access to pages through url inputs and search queries and element selection further allows navigation through the selection of hyperlinks.

Eye typing presents a broader challenge for many gaze-based applications beyond web browsing but remains essential. Countless gaze-based text entry methods have been proposed, including dwell-based virtual keyboards, gaze-gesture typing, and multimodal approaches combining gaze with voice or other inputs [15, 22, 32, 37].

Element selection remains a particularly complex issue due to the small size of elements relative to gaze-tracking precision, leading to potential ambiguity when multiple elements fall within the selection area, hence requiring assistance techniques [28]. Additionally, elements such as hyperlinks are often embedded within textual content or images, exacerbating the Midas Touch problem, as they are difficult to avoid gazing unintentionally.

A straightforward approach to gaze-based selection is direct pointing, in which a cursor moves to the observed location following a fixation, requiring a second fixation on a UI element to confirm selection (i.e., simulate a click) [2, 8]. However, this method is highly dependent on the accuracy of the eye tracker, often necessitating multiple fixations to correctly position the cursor, which can negatively impact efficiency and user experience.

Direct selection can be enhanced through magnification techniques, which enlarge the region of interest to increase the size of clickable elements, reducing ambiguity in selection. Magnification typically begins with the activation of a stable UI element of appropriate size and can be implemented in various ways. These include continuous magnification triggered by dwell time, discrete magnification that increases in fixed increments per fixation (in one or multiple steps), or localized magnification around the gaze point. The latter can be rendered with a sharp boundary, similar to a magnifying glass, or with a smooth transition effect, resembling a fisheye lens [3, 17, 29].

Another common approach is indirect pointing, where users gaze at a region of interest, and the system maps nearby clickable UI elements to larger selection targets, often on a separate, less cluttered area to mitigate involuntary selections. This technique, often referred to as the menu method, has been implemented in various ways, establishing a correspondence between original elements and selection buttons using connecting lines, color coding, numerical labels, or reproductions of the text and images from the original elements [8, 16, 27, 31, 35]. Another form of indirect pointing is discrete selection, where a selectable element is highlighted, and the user navigates through nearby elements using dedicated UI controls. These controls may include "previous" and "next" buttons, similar to tab navigation on a keyboard, or multidirectional arrows, moving to the nearest selectable element in that direction [8, 9].

Casarini, Porta, and Dondi [8] conducted a comparative study evaluating multiple hyperlink selection techniques, including menu-based selection, instant and progressive magnification methods, direct pointing, and discrete selection. Their findings indicate that menu-based selection results in faster selection times and higher user satisfaction compared to other methods.

Alternative approaches that do not rely on dwell-time have also been explored. Carneiro, Gonzales, and Morimoto [7] proposed a pursuit-based method in which users select links by tracking rotating targets positioned at the screen's edge, using their angular movement as an input signal. While this technique enables a higher density of selectable elements within a limited area, it comes at the cost of slower selection times compared to menu-based and magnification-based approaches.

Other tasks for browsing the web have also been explored as auxiliary features to the main hyperlinks selection. One of those is in-page navigation. Most implementations achieve this strictly through vertical scrolling, as most web pages are designed to fit horizontally. Scrolling mechanisms can be implicit—triggered by gaze location —or explicit, using dedicated gaze-activated buttons. Another form of navigation implemented is the zoom functionality, either by scaling the entire page or through temporary/local magnification effects [3, 16, 17, 29].

Another auxiliary feature is tab management. It is relatively straightforward tasks, as their user interface elements are predictable, discreet and can be constrained to mitigate issues such as precision limitations. While many options of click alternatives are available for tab management (dwell-time, gaze gestures etc.) few works have explored this functionality [29].

2.2 Gaze Interaction for XR

XR applications can leverage multiple input methods, including clickers, hand gestures, voice commands, and gaze-based interaction. Many applications adopt multimodal approaches, selecting the most effective input mode for each task or offering alternative options. For example, users may rely on voice input for text entry while having a gaze-based keyboard as an alternative when voice input is impractical. Gaze interaction is particularly valuable in XR environments due to its subtle nature, in contrast to more conspicuous methods. Additionally, it can help mitigate arm fatigue by serving as an alternative or complement to hand gestures or handheld controllers for pointing [18].

Gaze interaction has been shown to be effective in various XR scenarios. Steves, Shin, and Oakley [14] conducted a Fitts' Law study comparing different selection methods in augmented reality. Among the six techniques tested — device clicking, controller clicking, hand gestures, gaze-dwell, visual pursuits, and voice selection — the results indicated that controller clicking performed best in terms of selection time, error rate, and user preference, followed by gaze-dwell.

Gaze-based typing in XR has been extensively explored as a substitute for traditional keyboards. One approach involves using gaze to accelerate manual pointing with handheld or worn pointers. In desktop environments, gaze-assisted manual pointing is a well-established technique [40], where the cursor moves closer to the user's gaze before fine adjustments are made through a mouse. This method can be adapted to XR, enabling rapid transitions between distant keys while relying on manual input for precise selection, thereby improving typing speed and reducing arm fatigue [41].

Exclusive gaze-based typing in XR has also been investigated [25, 34, 39]. Traditional visible keyboards in XR can cause occlusion issues, prompting research into "invisible" keyboards that rely on alternative interaction techniques [23, 26, 42]. This concept has been extended to gaze typing by defining a typing region and leveraging users' spatial awareness of keyboard layouts to facilitate text entry without the need for a visible keyboard [24].

2.3 GazeBar

Despite gaze-dwell beeing the most common form of gaze selection mechanism it's not the only one. A particularly new method called GazeBar seeks to accelerate selection, and by consequence improve user experience. Introduced by Elmadjian and Morimoto [13], GazeBars were designed for a multimodal drawing application, where finer drawing controls were handled by a mouse or touch pad and switching between modes - pencil types, thickness, color, etc. - by gaze. By placing the bars at the edge of the screen, GazeBars can avoid indirect selections during regular use. Alongside that, changes made by different gaze bars buttons were not destructive, simply switching between modes, allowing for simple error corrections by switching the mode back. This approach permitted GazeBars to forego dwell times and perform instant selections when an item is glaces, allowing for a speed gain, at the cost of a slightly higher error rate. In the GazeBar method a selection is only done once the eyes cross the bar back into the main region, leaving the last looked item as selected. Although originally envisioned in a multimodal environment, GazeBars can be used in gaze only contexts

A GazeBar application has been adapted for XR environments by leveraging the natural tendency of eye gaze to avoid the very edges of the visual field. This allows interactive buttons to be positioned radially at the periphery of vision, minimizing interference with the main content while remaining easily accessible [38].

3 Design Proposal

Building on prior research, this paper presents an initial design for a gaze-based web browser tailored for XR environments in educational contexts, named `XRBars`.

`XRBars` adopts a traditional rectangular window layout, similar to desktop applications, as its initial development also targets desktop platforms. This does not compromise its usability in XR environments, where it can function as a floating window. Future developments might lead to a XR only version, taking advantage of the full scope of view, utilizing radial GazeBars.

To ensure a fully functional browsing experience, we define four key features:

- Page Interaction (Hyperlink access, UI element selection)
- Page navigation (panning and zooming)
- Tab management (creation, removal, bookmarking,searching, entering URLs)
- input (typing, filling forms, video controls)

Among these, hyperlink selection and text input have been the most extensively studied in gaze-based interaction research.

The proposed interface can be seen on figure 1, showcasing 3 bars corresponding to different features. The top bar is reserved for tab management, the right bar is for page navigation, finally the bottom bar is responsible for page interaction. Inputs will be handled through specific pop-ups dependent on their type.

Figure 1: Default state of `XRBars`. Author's Creation.

All shown bars are dwell free, meaning the moment the user glances at them the user can input commands without having to wait. The top and bottom bars are GazeBars, with discreet items that can be selected by returning the gaze to the center of the screen through their edge. The right bar is a 'GazeScroll' automatically following the user's gaze location with it's scroll bar.

3.1 Page Navigation

Page navigation is currently limited to scrolling, implemented through two complementary methods, which can be observed on Figure 2.

Gaze-activated scrolling: The page scrolls when the user gazes at invisible trigger zones at the top or bottom of the screen, as shown on the left side of figure 2 by striped bards, enabling smooth, continuous, fine-tuned scrolling—ideal for tasks such as reading.

Gaze-controlled scrollbar: A scrollbar positioned on the right side of the interface allows instant jumping to different sections of a page based on gaze position, visible in the right side of figure 2. While less precise than smooth scrolling, it facilitates rapid navigation of long pages.

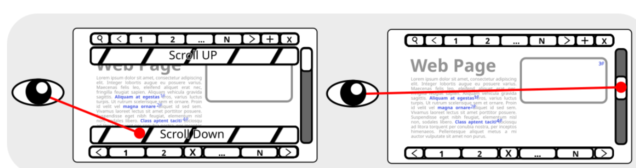

Figure 2: Two scroll types of `XRBars`: Smooth and Instant. Author's Creation.

By combining both methods, users can quickly locate content and then fine-tune their position as needed. Additional features, such as page zooming or local magnification, can be added via an additional optional to one of the existing bars or a new one at a latter moment if deemed necessary.

3.2 Tab Management

Tab management is handled by the top GazeBar, as in Figure 3, which displays a list of open tabs numbered 1 to N. The bar also includes navigation arrows for browsing additional tabs when space is limited, ensuring each tab retains appropriate size for selection for any number of tabs. Other controls include: a magnifying glass icon to open the URL search and entry menu; a '+' button to open a new tab and a 'X' button to close existing tabs, located near tab switching due to their semantically related functions.

Tabs are automatically selected when the user glances at them, enabling peripheral identification of the current selection. Similarly, looking at the navigation arrows scrolls through the tab list. However, commands that modify the user's workflow (e.g., opening/closing tabs) are only executed when the user glances away from the bar, preventing accidental activation.

Figure 3: Tab Management GazeBar. Author's Creation.

3.3 Page Interaction

The bottom GazeBar, showcased in Figure 4, is dedicated to hyperlink selection, form interactions, text input access and other forms of page element interactions. Interaction follows a three-step process. As the user browses the page, the system dynamically labels interactive elements near their region of interest with numbered tags, as shown in the left-hand side of Figure 4. When ready to select an item, the user glances at the GazeBar, where buttons corresponding to the numbered elements appear, along with a preview of it's interaction type, link access, form field, image preview etc, seen on the right-hand side of Figure 4. Once the user find the desired element they confirm the selection by crossing their gaze over the desired option back into the central region.

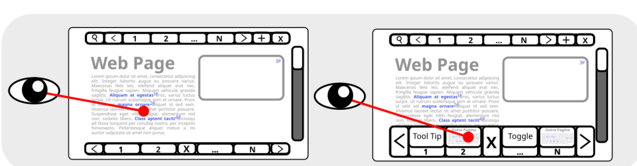

Figure 4: Page Interaction GazeBar in use, on the left, the initial labeling, on the right, element selection. Author's Creation.

Some interactions types are concluded there, such as hyperlink selection, which would redirect the user to a new page, but each element type interacts in a distinct manner.

This approach ensures that small, difficult-to-target elements are mapped to larger, easily selectable buttons, streamlining interaction. Since GazeBars trigger actions when gaze crosses regions, a cancel ('X') button is provided as a safe area to disengage without unintended activation.

At the far ends of the GazeBar, page forward and back buttons enable navigation between previously accessed pages. These controls are grouped here as they are closely related to hyperlink selection and cross-page navigation, as well as means of error correcting in case a wrong hyperlink is selected.

3.4 Input

There are several input types a browser might expect, toggle clicking, text typing, video controls etc. At this stage, the design does not prescribe a specific solution to most of those input types. In particular, we highlight hyperlinks and toggle elements, which trigger instantly without pop-ups, and redirect and change states, respectively. Despite their simplicity, these elements combined serve as a robust framework for multiple-selection exams to be held with this system, using toggles as options and hyperlinks as confirmation.

4 Experiment

As a means of evaluating `XRBars`, a within-subjects usability experiment will be conducted, with the browser used as the sole factor: `XRBars` and another state-of-the-art gaze-based browser. The experiment will measure completion time, as a means of assessing efficiency, and cognitive load, as a means of evaluating interaction fluidity.

The task will involve filling out a multiple-choice test with 20 questions, using answers found on a separate tab. This task ensures that the primary features—selection, tab switching, and scrolling—are exercised.

Participants will be instructed to complete the test by switching between tabs as needed and to click the Done button at the bottom of the page once finished.

User data will be recorded in the form of screen recordings with gaze coordinates, to further illuminate user behavior—for example, identifying which interface prompted more tab switching, or how frequently participants revised previous answers.

5 Conclusion

This papers described `XRBarss`, a gaze-based interface to improve the speed and reliability of selections by gaze in web browsers. `XRBarss` is an extension of the GazeBars concept, initially designed for multi-modal interaction in desktop computers. With the emergence of wearable XR platforms such as the Apple's IVision Pro, novel interfaces such as `XRBars` can improve the life of people with motor disabilities, facilitating their access to the Internet and, in particular, the access to many learning tools and applications. Future research will include the implementation of `XRBars` and it's integration in an open source browser, such as Chromium, as well as it's evaluation in terms of of usability and application for learning platforms, such as evaluating tools composed of multiple choice questions, that can be easily extended to other types of assessment questions, such as true/false and associative questions.

References

[1] Colin Allison, Alan Miller, Iain Oliver, Rosa Michaelson, and Thanassis Tiropanis. 2012. The Web in education. *Computer Networks* 56, 18 (2012), 3811–3824.

[2] Emmanuel Arias, Gustavo López, Luis Quesada, and Luis Guerrero. 2016. Web accessibility for people with reduced mobility: a case study using eye tracking. In *Advances in Design for Inclusion: Proceedings of the AHFE 2016 International Conference on Design for Inclusion, July 27-31, 2016, Walt Disney World®, Florida, USA*. Springer, 463–473.

[3] Michael Ashmore, Andrew T Duchowski, and Garth Shoemaker. 2005. Efficient eye pointing with a fisheye lens. In *Proceedings of Graphics interface 2005*. 203–210.

[4] Dennis Beck. 2019. Augmented and virtual reality in education: Immersive learning research. *Journal of Educational Computing Research* 57, 7 (2019), 1619–1625.

[5] Peter Brophy and Jenny Craven. 2007. Web accessibility. *Library trends* 55, 4 (2007), 950–972.

[6] Milton Campoverde-Molina, Sergio Lujan-Mora, and Llorenc Valverde Garcia. 2020. Empirical studies on web accessibility of educational websites: A systematic literature review. *IEEE Access* 8 (2020), 91676–91700.

[7] Alex Torquato Souza Carneiro, Candy Veronica Tenorio Gonzales, and Carlos Hitoshi Morimoto. 2023. EyePursuitLinks - an Eye-pursuit Based Interface for Web Browsing Using Smart Targets. In *Proceedings of the 29th Brazilian Symposium on Multimedia and the Web* (, Ribeirão Preto, Brazil,) *(WebMedia '23)*. Association for Computing Machinery, New York, NY, USA, 16–24. doi:10.1145/3617023.3617058

[8] Matteo Casarini, Marco Porta, and Piercarlo Dondi. 2020. A gaze-based web browser with multiple methods for link selection. In *ACM Symposium on Eye Tracking Research and Applications*. 1–8.

[9] Emiliano Castellina, Fulvio Corno, et al. 2007. Accessible web surfing through gaze interaction. In *Gaze-based Creativity, Interacting with Games and On-line Communities*. 74–77.

[10] Raphael Cohen-Almagor. 2013. Internet history. In *Moral, ethical, and social dilemmas in the age of technology: Theories and practice*. IGI Global, 19–39.

[11] Inmaculada Coma-Tatay, Sergio Casas-Yrurzum, Pablo Casanova-Salas, and Marcos Fernández-Marín. 2019. FI-AR learning: a web-based platform for augmented reality educational content. *Multimedia Tools and Applications* 78, 5 (2019), 6093–6118.

[12] Piercarlo Dondi and Marco Porta. 2023. Gaze-based human–computer interaction for museums and exhibitions: technologies, Applications and Future Perspectives. *Electronics* 12, 14 (2023), 3064.

[13] Carlos Elmadjian and Carlos H Morimoto. 2021. Gazebar: Exploiting the midas touch in gaze interaction. In *Extended abstracts of the 2021 CHI conference on human factors in computing systems*. 1–7.

[14] Augusto Esteves, Yonghwan Shin, and Ian Oakley. 2020. Comparing selection mechanisms for gaze input techniques in head-mounted displays. *International Journal of Human-Computer Studies* 139 (2020), 102414.

[15] Wenxin Feng, Jiangnan Zou, Andrew Kurauchi, Carlos H Morimoto, and Margrit Betke. 2021. *HGaze Typing: Head-Gesture Assisted Gaze Typing*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3448017.3457379

[16] Pedro Figueiredo and Manuel J Fonseca. 2018. EyeLinks: a gaze-only click alternative for heterogeneous clickables. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*. 307–314.

[17] Tovi Grossman and Ravin Balakrishnan. 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 281–290.

[18] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1063–1072.

[19] Dave J Hobbs and RJ Taylor. 1996. The Impact on Education of the World Wide Web. ERIC.

[20] Robert JK Jacob. 1995. Eye tracking in advanced interface design. *Virtual environments and advanced interface design* 258, 288 (1995), 2.

[21] Mohammad Amin Kuhail, Areej ElSayary, Shahbano Farooq, and Ahlam Alghamdi. 2022. Exploring immersive learning experiences: A survey. In *Informatics*, Vol. 9. MDPI, 75.

[22] Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos H. Morimoto, and Margrit Betke. 2020. Swipe&Switch: Text Entry Using Gaze Paths and Context Switching. In *Adjunct Publication of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 84–86. doi:10.1145/3379350.3416193

[23] Lik Hang Lee, Kit Yung Lam, Yui Pan Yau, Tristan Braud, and Pan Hui. 2019. Hibey: Hide the keyboard in augmented reality. In *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom*. IEEE, 1–10.

[24] Xueshi Lu, Difeng Yu, Hai-Ning Liang, and Jorge Goncalves. 2021. itext: Hands-free text entry on an imaginary keyboard for augmented reality systems. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 815–825.

[25] Xueshi Lu, Difeng Yu, Hai-Ning Liang, Wenge Xu, Yuzheng Chen, Xiang Li, and Khalad Hasan. 2020. Exploration of hands-free text entry techniques for virtual reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 344–349.

[26] Yiqin Lu, Chun Yu, Xin Yi, Yuanchun Shi, and Shengdong Zhao. 2017. Blind-type: Eyes-free text entry on handheld touchpad by leveraging thumb's muscle memory. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 1–24.

[27] Christof Lutteroth, Moiz Penkar, and Gerald Weber. 2015. Gaze vs. mouse: A fast and accurate gaze-only click alternative. In *Proceedings of the 28th annual ACM symposium on user interface software & technology*. 385–394.

[28] I Scott MacKenzie. 2010. An eye on input: research challenges in using the eye for computer input control. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. 11–12.

[29] Raphael Menges, Chandan Kumar, and Steffen Staab. 2019. Improving User Experience of Eye Tracking-Based Interaction: Introspecting and Adapting Interfaces. In *ACM Trans. Comput.-Hum. Interact.*, Vol. 26. ACM, New York, NY, USA, Article 37, 46 pages. doi:10.1145/3338844

[30] Shailendra Palvia, Prageet Aeron, Parul Gupta, Diptiranjan Mahapatra, Ratri Parida, Rebecca Rosner, and Sumita Sindhi. 2018. Online education: Worldwide status, challenges, trends, and implications. 233–241 pages.

[31] Abdul Moiz Penkar, Christof Lutteroth, and Gerald Weber. 2013. Eyes only: Navigating hypertext with gaze. In *Human-Computer Interaction–INTERACT 2013: 14th IFIP TC 13 International Conference, Cape Town, South Africa, September 2-6, 2013, Proceedings, Part II 14*. Springer, 153–169.

[32] Marco Porta and Matteo Turina. 2008. Eye-S: a full-screen input modality for pure eye-based communication. In *Proceedings of the 2008 symposium on Eye tracking research & applications*. 27–34.

[33] Xiuquan Qiao, Pei Ren, Schahram Dustdar, Ling Liu, Huadong Ma, and Junliang Chen. 2019. Web AR: A promising future for mobile augmented reality—State of the art, challenges, and insights. *Proc. IEEE* 107, 4 (2019), 651–666.

[34] Vijay Rajanna and John Paulin Hansen. 2018. Gaze typing in virtual reality: impact of keyboard design, selection method, and motion. In *Proceedings of the 2018 ACM symposium on eye tracking research & applications*. 1–10.

[35] Daniel Vella and Chris Porter. 2024. Remapping the document object model using geometric and hierarchical data structures for efficient eye control. *Proceedings of the ACM on Human-Computer Interaction* 8, ETRA (2024), 1–16.

[36] Zhimin Wang, Maohang Rao, Shanghua Ye, Weitao Song, and Feng Lu. 2025. Towards spatial computing: recent advances in multimodal natural interaction for XR headsets. *arXiv preprint arXiv:2502.07598* (2025).

[37] Zhimin Wang, Maohang Rao, Shanghua Ye, Weitao Song, and Feng Lu. 2025. Towards spatial computing: recent advances in multimodal natural interaction for XR headsets. *arXiv preprint arXiv:2502.07598* (2025).

[38] Xin Yi, Yiqin Lu, Ziyin Cai, Zihan Wu, Yuntao Wang, and Yuanchun Shi. 2022. Gazedock: Gaze-only menu selection in virtual reality using auto-triggering peripheral menu. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 832–842.

[39] Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. 2017. Tap, dwell or gesture? exploring head-based text entry techniques for hmds. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 4479–4488.

[40] Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 246–253.

[41] Maozheng Zhao, Alec M Pierce, Ran Tan, Ting Zhang, Tianyi Wang, Tanya R Jonker, Hrvoje Benko, and Aakar Gupta. 2023. Gaze speedup: Eye gaze assisted gesture typing in virtual reality. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*. 595–606.

[42] Suwen Zhu, Tianyao Luo, Xiaojun Bi, and Shumin Zhai. 2018. Typing on an invisible keyboard. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.